The problem of ontology alignment on the web: a first report

Davide Fossati and **Gabriele Ghidoni** and **Barbara Di Eugenio**
and **Isabel Cruz** and **Huiyong Xiao** and **Rajen Subba**
Computer Science
University of Illinois
Chicago, IL, USA
dfossa1@uic.edu, red.one.999@virgilio.it, bdieugen@cs.uic.edu
ifc@cs.uic.edu, hxiao2@uic.edu, rsubba@cs.uic.edu

Abstract

This paper presents a general architecture and four algorithms that use Natural Language Processing for automatic ontology matching. The proposed approach is purely instance based, i.e., only the instance documents associated with the nodes of ontologies are taken into account. The four algorithms have been evaluated using real world test data, taken from the Google and LookSmart online directories. The results show that NLP techniques applied to instance documents help the system achieve higher performance.

1 Introduction

Many fundamental issues about the viability and exploitation of the web as a linguistic corpus have not been tackled yet. The web is a massive repository of text and multimedia data. However, there is not a systematic way of classifying and retrieving these documents. Computational Linguists are of course not the only ones looking at these issues; research on the Semantic Web focuses on providing a semantic description of all the resources on the web, resulting into *a mesh of information linked up in such a way as to be easily processable by machines, on a global scale. You can think of it as being an efficient way of representing data on the World Wide Web, or as a globally linked database.*[1] The way the vision of the Semantic Web will be achieved, is by describing each document using languages such as RDF Schema and OWL, which are capable of explicitly expressing the meaning of terms in vocabularies and the relationships between those terms.

[1] `http://infomesh.net/2001/swintro/`

The issue we are focusing on in this paper is that these languages are used to define ontologies as well. If ultimately a single ontology were used to describe all the documents on the web, systems would be able to exchange information in a transparent way for the end user. The availability of such a standard ontology would be extremely helpful to NLP as well, e.g., it would make it far easier to retrieve all documents on a certain topic. However, until this vision becomes a reality, a plurality of ontologies are being used to describe documents and their content. The task of *automatic ontology alignment* or *matching* (Hughes and Ashpole, 2005) then needs to be addressed.

The task of ontology matching has been typically carried out manually or semi-automatically, for example through the use of graphical user interfaces (Noy and Musen, 2000). Previous work has been done to provide automated support to this time consuming task (Rahm and Bernstein, 2001; Cruz and Rajendran, 2003; Doan et al., 2003; Cruz et al., 2004; Subba and Masud, 2004). The various methods can be classified into two main categories: *schema based* and *instance based*. *Schema based* approaches try to infer the semantic mappings by exploiting information related to the structure of the ontologies to be matched, like their topological properties, the labels or description of their nodes, and structural constraints defined on the schemas of the ontologies. These methods do not take into account the actual data classified by the ontologies. On the other hand, *instance based* approaches look at the information contained in the instances of each element of the schema. These methods try to infer the relationships between the nodes of the ontologies from the analysis of their instances. Finally, *hybrid* approaches combine schema and instance based

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The problem of ontology alignment on the web: a first report	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Illinois,Department of Computer Science ,Urbana,IL,61801**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**8**	

methods into integrated systems.

Neither instance level information, nor NLP techniques have been extensively explored in previous work on ontology matching. For example, (Agirre et al., 2000) exploits documents (instances) on the WWW to enrich WordNet (Miller et al., 1990), i.e., to compute "concept signatures," collection of words that significantly distinguish one sense from another, however, not directly for ontology matching. (Liu et al., 2005) uses documents retrieved via queries augmented with, for example, synonyms that WordNet provides to improve the accuracy of the queries themselves, but not for ontology matching. NLP techniques such as POS tagging, or parsing, have been used for ontology matching, but on the names and definitions in the ontology itself, for example, in (Hovy, 2002), hence with a schema based methodology.

In this paper, we describe the results we obtained when using some simple but effective NLP methods to align web ontologies, using an instance based approach. As we will see, our results show that more sophisticated methods do not necessarily lead to better results.

2 General architecture

The instance based approach we propose uses NLP techniques to compute matching scores based on the documents classified under the nodes of ontologies. There is no assumption on the structural properties of the ontologies to be compared: they can be any kind of graph representable in OWL. The instance documents are assumed to be text documents (plain text or HTML).

The matching process starts from a pair of ontologies to be aligned. The two ontologies are traversed and, for each node having at least one instance, the system computes a *signature* based on the instance documents. Then, the signatures associated to the nodes of the two ontologies are compared pairwise, and a similarity score for each pair is generated. This score could then be used to estimate the likelihood of a match between a pair of nodes, under the assumption that the semantics of a node corresponds to the semantics of the instance documents classified under that node. Figure 1 shows the architecture of our system.

The two main issues to be addressed are (1) the representation of signatures and (2) the definition of a suitable comparison metric between signatures. For a long time, the Information Retrieval community has succesfully adopted a "bag of words" approach to effectively represent and compare text documents. We start from there to define a general signature structure and a metric to compare signatures.

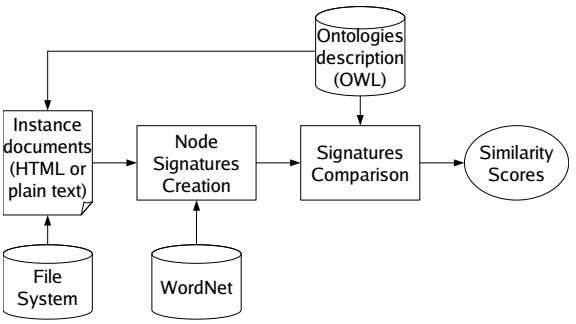

Figure 1: Ontology aligment architecture

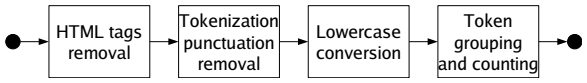

Figure 2: Baseline signature creation

A *signature* is defined as a function $S : K \rightarrow R^+$, mapping a finite set of *keys* (which can be complex objects) to positive real values. With a signature of that form, we can use the *cosine similarity* metric to score the similarity between two signatures:

$$simil(S_1, S_2) = \frac{\sum_p S_1(k_p) S_2(k_p)}{\sqrt{\sum_i S_1(k_i)^2} \cdot \sqrt{\sum_j S_2(k_j)^2}}$$

$$k_p \in K_1 \cap K_2,\ k_i \in K_1,\ k_j \in K_2$$

The cosine similarity formula produces a value in the range [0, 1]. The meaning of that value depends on the algorithm used to build the signature. In particular, there is no predefined threshold that can be used to discriminate matches from non-matches. However, such a threshold could be computed a-posteriori from a statistical analysis of experimental results.

2.1 Signature generation algorithms

For our experiments, we defined and implemented four algorithms to generate signatures. The four algorithms make use of text and language processing techniques of increasing complexity.

2.1.1 Algorithm 1: Baseline signature

The baseline algorithm performs a very simple sequence of text processing, schematically represented in Figure 2.

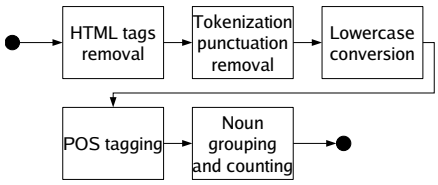

Figure 3: Noun signature creation

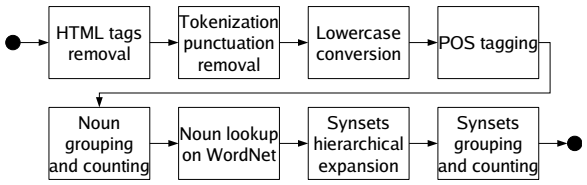

Figure 4: WordNet signature creation

HTML tags are first removed from the instance documents. Then, the texts are tokenized and punctuation is removed. Everything is then converted to lowercase. Finally, the tokens are grouped and counted. The final signature has the form of a mapping table *token* → *frequency*.

The main problem we expected with this method is the presence of a lot of noise. In fact, many "irrelevant" words, like determiners, prepositions, and so on, are added to the final signature.

2.1.2 Algorithm 2: Noun signature

To cope with the problem of excessive noise, people in IR often use fixed lists of *stop words* to be removed from the texts. Instead, we introduced a syntax based filter in our chain of processing. The main assuption is that nouns are the words that carry most of the meaning for our kind of document comparison. Thus, we introduced a part-of-speech tagger right after the tokenization module (Figure 3). The results of the tagger are used to discard everything but nouns from the input documents. The part-of-speech tagger we used –QTAG 3.1 (Tufis and Mason, 1998), readily available on the web as a Java library– is a Hidden Markov Model based statistical tagger.

The problems we expected with this approach are related to the high specialization of words in natural language. Different nouns can bear similar meaning, but our system would treat them as if they were completely unrelated words. For example, the words "apple" and "orange" are semantically closer than "apple" and "chair," but a purely syntactic approach would not make any difference between these two pairs. Also, the current method does not include morphological processing, so different inflections of the same word, such as "apple" and "apples," are treated as distinct words.

In further experiments, we also considered verbs, another syntactic category of words bearing a lot of semantics in natural language. We computed signatures with verbs only, and with verbs and nouns together. In both cases, however, the performance of the system was worse. Thus, we will not consider verbs in the rest of the paper.

2.1.3 Algorithm 3: WordNet signature

To address the limitations stated above, we used the WordNet lexical resource (Miller et al., 1990). WordNet is a dictionary where words are linked together by semantic relationships. In WordNet, words are grouped into *synsets*, i.e., sets of synonyms. Each synset can have links to other synsets. These links represent semantic relationships like hypernymy, hyponymy, and so on.

In our approach, after the extraction of nouns and their grouping, each noun is looked up on WordNet (Figure 4). The synsets to which the noun belongs are added to the final signature in place of the noun itself. The signature can also be enriched with the hypernyms of these synsets, up to a specified level. The final signature has the form of a mapping *synset* → *value*, where *value* is a weighted sum of all the synsets found.

Two important parameters of this method are related to the hypernym expansion process mentioned above. The first parameter is the maximum level of hypernyms to be added to the signature (*hypernym level*). A hypernym level value of 0 would make the algorithm add only the synsets of a word, without any hypernym, to the signature. A value of 1 would cause the algorithm to add also their parents in the hypernym hierarchy to the signature. With higher values, all the ancestors up to the specified level are added. The second parameter, *hypernym factor*, specifies the damping of the weight of the hypernyms in the expansion process. Our algorithm exponentially dampens the hypernyms, i.e., the weigth of a hypernym decreases exponentially as its level increases. The hypernym factor is the base of the exponential function.

In general, a noun can have more than one sense, e.g., "apple" can be either a fruit or a tree. This is reflected in WordNet by the fact that a noun can belong to multiple synsets. With the current approach, the system cannot decide which

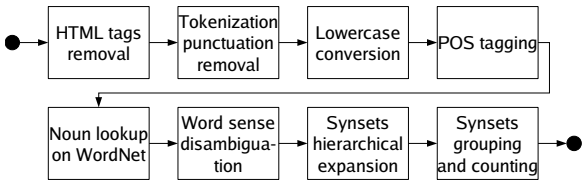

Figure 5: Disambiguated signature creation

sense is the most appropriate, so all the senses of a word are added to the final signature, with a weight inversely proportional to the number of possible senses of that word. This fact potentially introduces *semantic noise* in the signature, because many irrelevant senses might be added to the signature itself.

Another limitation is that a portion of the nouns in the source texts cannot be located in WordNet (see Figure 6). Thus, we also tried a variation (algorithm 3+2) that falls back on to the bare lexical form of a noun if it cannot be found in WordNet. This variation, however, resulted in a slight decrease of performance.

2.1.4 Algorithm 4: Disambiguated signature

The problem of having multiple senses for each word calls for the adoption of word sense disambiguation techniques. Thus, we implemented a word sense disambiguator algorithm, and we inserted it into the signature generation pipeline (Figure 5). For each noun in the input documents, the disambiguator takes into account a specified number of *context words*, i.e., nouns preceding and/or following the target word. The algorithm computes a measure of the *semantic distance* between the possible senses of the target word and the senses of each of its context words, pairwise. A sense for the target word is chosen such that the total distance to its context is minimized. The *semantic distance* between two synsets is defined here as the minimum number of hops in the WordNet hypernym hierarchy connecting the two synsets. This definition allows for a relatively straightforward computation of the semantic distance using WordNet. Other more sophisticated definitions of semantic distance can be found in (Patwardhan et al., 2003). The word sense disambiguation algorithm we implemented is certainly simpler than others proposed in the literature, but we used it to see whether a method that is relatively simple to implement could still help.

The overall parameters for this signature creation algorithm are the same as the WordNet signature algorithm, plus two additional parameters for the word sense disambiguator: *left context length* and *right context length*. They represent respectively how many nouns before and after the target should be taken into account by the disambiguator. If those two parameters are both set to zero, then no context is provided, and the first possible sense is chosen. Notice that even in this case the behaviour of this signature generation algorithm is different from the previous one. In a WordNet signature, every possible sense for a word is inserted, whereas in a WordNet disambiguated signature only one sense is added.

3 Experimental setting

All the algorithms described in the previous section have been fully implemented in a coherent and extensible framework using the Java programming language, and evaluation experiments have been run. This section describes how the experiments have been conducted.

3.1 Test data

The evaluation of ontology matching approaches is usually made difficult by the scarceness of test ontologies readily available in the community. This problem is even worse for instance based approaches, because the test ontologies need also to be "filled" with instance documents. Also, we wanted to test our algorithms with "real world" data, rather than toy examples.

We were able to collect suitable test data starting from the ontologies published by the Ontology Alignment Evaluation Initiative 2005 (Euzenat et al., 2005). A section of their data contained an OWL representation of fragments of the Google, Yahoo, and LookSmart web directories. We "reverse engineered" some of this fragments, in order to reconstruct two consistent trees, one representing part of the Google directory structure, the other representing part of the LookSmart hierarchy. The leaf nodes of these trees were filled with instances downloaded from the web pages classified by the appropriate directories. With this method, we were able to fill 7 nodes of each ontology with 10 documents per node, for a total of 140 documents. Each document came from a distinct web page, so there was no overlap in the data to be compared. A graphical representation of our two test ontologies, *source* and *target*, is shown in Fig-

ure 6. The darker outlined nodes are those filled with instance documents. For the sake of readability, the names of the nodes corresponding to real matches are the same. Of course, this information is not used by our algorithms, which adopt a purely instance based approach. Figure 6 also reports the size of the instance documents associated to each node: total number of words, noun tokens, nouns, and nouns covered by WordNet.

3.2 Parameters

The experiments have been run with several combinations of the relevant parameters: number of instance documents per node (5 or 10), algorithm (1 to 4), extracted parts of speech (nouns, verbs, or both), hypernym level (an integer value equal or greater than zero), hypernym factor (a real number), and context length (an integer number equal or greater than zero). Not all of the parameters are applicable to every algorithm. The total number of runs was 90.

4 Results

Each run of the system with our test ontologies produced a set of 49 values, representing the matching score of every pair of nodes containing instances across the two ontologies. Selected examples of these results are shown in Tables 1, 2, 3, and 4. In the experiments shown in those tables, 10 instance documents for each node were used to compute the signatures. Nodes that actually match (identified by the same label, e.g., "Canada" and "Canada") should show high similarity scores, whereas nodes that do not match (e.g., "Canada" and "Dendrochronology"), should have low scores. Better algorithms would have higher scores for matching nodes, and lower score for non-matching ones. Notice that the two nodes "Egypt" and "Pyramid Theories," although intuitively related, have documents that take different perspectives on the subject. So, the algorithms correctly identify the nodes as being different.

Looking at the results in this form makes it difficult to precisely assess the quality of the algorithms. To do so, a statistical analysis has to be performed. For each table of results, let us partition the scores in two distinct sets:

$$A = \{simil(node_i, node_j) \mid \text{real match} = true\}$$

$$B = \{simil(node_i, node_j) \mid \text{real match} = false\}$$

	Target node						
Source node	Canada	Dendro chronology	Megaliths	Museums	Nazca Lines	Pyramid Theories	United Kingdom
Canada	**0.95**	0.89	0.89	0.91	0.87	0.86	0.92
Dendro chronology	0.90	**0.97**	0.91	0.90	0.88	0.87	0.92
Egypt	0.86	0.89	0.91	0.87	0.86	0.88	0.90
Megaliths	0.90	0.91	**0.99**	0.93	0.95	0.94	0.93
Museums	0.89	0.88	0.90	**0.93**	0.88	0.87	0.90
Nazca Lines	0.88	0.88	0.95	0.91	**0.99**	0.93	0.91
United Kingdom	0.87	0.87	0.86	0.88	0.82	0.82	**0.96**

Table 1: Results – Baseline signature algorithm

	Target node						
Source node	Canada	Dendro chronology	Megaliths	Museums	Nazca Lines	Pyramid Theories	United Kingdom
Canada	**0.67**	0.20	0.14	0.35	0.08	0.08	0.41
Dendro chronology	0.22	**0.80**	0.15	0.22	0.09	0.09	0.25
Egypt	0.13	0.23	0.26	0.22	0.17	0.24	0.25
Megaliths	0.28	0.20	**0.85**	0.37	0.22	0.27	0.33
Museums	0.30	0.19	0.18	**0.58**	0.08	0.14	0.27
Nazca Lines	0.13	0.12	0.26	0.18	**0.96**	0.14	0.17
United Kingdom	0.42	0.20	0.17	0.26	0.09	0.11	**0.80**

Table 2: Results – Noun signature algorithm

	Target node						
Source node	Canada	Dendro chronology	Megaliths	Museums	Nazca Lines	Pyramid Theories	United Kingdom
Canada	**0.79**	0.19	0.19	0.38	0.15	0.06	0.56
Dendro chronology	0.26	**0.83**	0.18	0.20	0.16	0.07	0.24
Egypt	0.17	0.24	0.32	0.21	0.31	0.30	0.27
Megaliths	0.39	0.21	**0.81**	0.41	0.40	0.25	0.42
Museums	0.31	0.14	0.17	**0.70**	0.11	0.11	0.26
Nazca Lines	0.24	0.20	0.42	0.29	**0.91**	0.21	0.29
United Kingdom	0.56	0.17	0.22	0.25	0.15	0.08	**0.84**

Table 3: Results – WordNet signature algorithm (hypernym level=0)

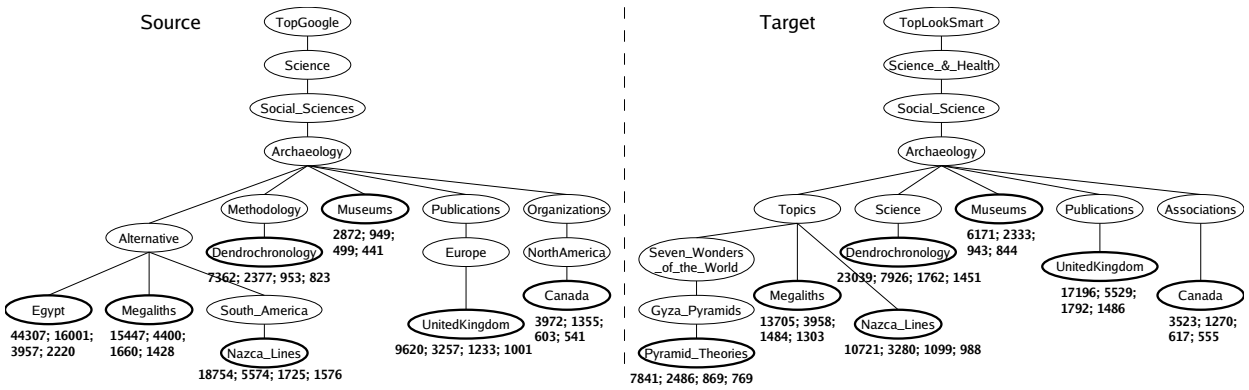

Figure 6: Ontologies used in the experiments. The numbers below the leaves indicate the size of instance documents: # of words; # of noun tokens; # of nouns; # of nouns in WordNet

	Target node						
Source node	Canada	Dendro chronology	Mega liths	Muse ums	Nazca Lines	Pyramid Theories	United Kingdom
Canada	**0.68**	0.18	0.13	0.33	0.12	0.05	0.44
Dendro chronology	0.23	**0.79**	0.15	0.20	0.14	0.07	0.23
Egypt	0.15	0.23	0.28	0.22	0.27	0.31	0.27
Megaliths	0.30	0.18	**0.84**	0.37	0.34	0.27	0.33
Museums	0.29	0.16	0.15	**0.60**	0.11	0.10	0.24
Nazca Lines	0.20	0.17	0.38	0.26	**0.89**	0.21	0.26
United Kingdom	0.45	0.17	0.18	0.24	0.15	0.08	**0.80**

Table 4: Results – Disambiguated signature algorithm (hypernym level=0, left context=1, right context=1)

With our test data, we would have 6 values in set A and 43 values in set B. Then, let us compute average and standard deviation of the values included in each set. The average of A represents the expected score that the system would assign to a match; likewise, the average of B is the expected score of a non-match. We define the following measure to compare the performance of our matching algorithms, inspired by "effect size" from (VanLehn et al., 2005):

$$\text{discrimination size} = \frac{avg(A) - avg(B)}{stdev(A) + stdev(B)}$$

Higher discrimination values mean that the scores assigned to matches and non-matches are more "far away," making it possible to use those scores to make more reliable decisions about the matching degree of pairs of nodes.

Table 5 shows the values of discrimination size (last column) out of selected results from our experiments. The algorithm used is reported in the first column, and the values of the other relevant parameters are indicated in other columns. We can make the following observations.

- Algorithms 2, 3, and 4 generally outperform the baseline (algorithm 1).
- Algorithm 2 (Noun signature), which still uses a fairly simple and purely syntactical technique, shows a substantial improvement. Algorithm 3 (WordNet signature), which introduces some additional level of semantics, has even better performance.
- In algorithms 3 and 4, hypernym expansion looks detrimental to performance. In fact, the best results are obtained with hypernym level equal to zero (no hypernym expansion).
- The word sense disambiguator implemented in algorithm 4 does not help. Even though disambiguating with some limited context (1 word before and 1 word after) provides slightly better results than choosing the first available sense for a word (context length equal to zero), the overall results are worse than adding all the possible senses to the signature (algorithm 3).
- Using only 5 documents per node significantly degrades the performance of all the algorithms (see the last 5 lines of the table).

5 Conclusions and future work

The results of our experiments point out several research questions and directions for future work,

Alg	Docs	POS	Hyp lev	Hyp fac	L cont	R cont	Avg (A)	Stdev (A)	Avg (B)	Stdev (B)	Discrimination size
1	10						0.96	0.02	0.89	0.03	**1.37**
2	10	noun					0.78	0.13	0.21	0.09	**2.55**
2	10	verb					0.64	0.20	0.31	0.11	1.04
2	10	nn+vb					0.77	0.14	0.21	0.09	2.48
3	10	noun	0				0.81	0.07	0.25	0.12	**3.08**
3	10	noun	1	1			0.85	0.07	0.41	0.12	2.35
3	10	noun	1	2			0.84	0.07	0.34	0.12	2.64
3	10	noun	1	3			0.83	0.07	0.31	0.12	2.80
3	10	noun	2	1			0.90	0.06	0.62	0.11	1.64
3	10	noun	2	2			0.86	0.07	0.45	0.12	2.18
3	10	noun	2	3			0.84	0.07	0.36	0.12	2.56
3	10	noun	3	1			0.95	0.04	0.78	0.08	1.44
3	10	noun	3	2			0.88	0.07	0.52	0.12	1.91
3	10	noun	3	3			0.85	0.07	0.38	0.12	2.45
3+2	10	noun	0	0			0.80	0.09	0.21	0.11	**2.94**
3+2	10	noun	1	2			0.83	0.08	0.30	0.11	2.73
3+2	10	noun	2	2			0.85	0.08	0.39	0.11	2.40
4	10	noun	0		0	0	0.80	0.12	0.24	0.10	2.64
4	10	noun	0		1	1	0.77	0.11	0.22	0.10	**2.67**
4	10	noun	0		2	2	0.77	0.11	0.23	0.10	2.59
4	10	noun	1	2	0	0	0.82	0.10	0.29	0.10	2.56
4	10	noun	1	2	1	1	0.80	0.10	0.34	0.10	2.27
4	10	noun	1	2	2	2	0.80	0.10	0.35	0.10	2.22
1	5	noun					0.93	0.05	0.86	0.04	0.88
2	5	noun					0.66	0.23	0.17	0.08	1.61
3	5	noun	0				0.70	0.17	0.21	0.11	**1.76**
4	5	noun	0		0	0	0.69	0.21	0.20	0.09	1.63
4	5	noun	0		1	1	0.64	0.21	0.18	0.08	1.58

Table 5: Results – Discrimination size

some more specific and some more general. As regards the more specific issues,

- Algorithm 2 does not perform morphological processing, whereas Algorithm 3 does. How much of the improved effectiveness of Algorithm 3 is due to this fact? To answer this question, Algorithm 2 could be enhanced to include a morphological processor.

- The effectiveness of Algorithms 3 and 4 may be hindered by the fact that many words are not yet included in the WordNet database (see Figure 6). Falling back on to Algorithm 2 proved not to be a solution. The impact of the incompleteness of the lexical resource should be investigated and assessed more precisely. Another venue of research may be to exploit different thesauri, such as the ones automatically derived as in (Curran and Moens, 2002).

- The performance of Algorithm 4 might be improved by using more sophisticated word sense disambiguation methods. It would also be interesting to explore the application of the unsupervised method described in (McCarthy et al., 2004).

As regards our long term plans, first, structural properties of the ontologies could potentially be exploited for the computation of node signatures. This kind of enhancement would make our system move from a purely instance based approach to a combined hybrid approach based on schema and instances.

More fundamentally, we need to address the lack of appropriate, domain specific resources that can support the training of algorithms and models appropriate for the task at hand. WordNet is a very general lexicon that does not support domain specific vocabulary, such as that used in geosciences or in medicine or simply that contained in a subontology that users may define according to their interests. Of course, we do not want to develop by hand domain specific resources that we have to change each time a new domain arises.

The crucial research issue is how to exploit extremely scarce resources to build efficient and effective models. The issue of scarce resources makes it impossible to use methods that are succesful at discriminating documents based on the words they contain but that need large corpora for training, for example Latent Semantic Analysis (Landauer et al., 1998). The experiments described in this paper could be seen as providing

a *bootstrapped* model (Riloff and Jones, 1999; Ng and Cardie, 2003)—in ML, bootstrapping requires to seed the classifier with a small number of well chosen target examples. We could develop a web spider, based on the work described on this paper, to automatically retrieve larger amounts of training and test data, that in turn could be processed with more sophisticated NLP techniques.

Acknowledgements

This work was partially supported by NSF Awards IIS–0133123, IIS–0326284, IIS–0513553, and ONR Grant N00014–00–1–0640.

References

Eneko Agirre, Olatz Ansa, Eduard Hovy, and David Martinez. 2000. Enriching very large ontologies using the WWW. In *ECAI Workshop on Ontology Learning*, Berlin, August.

Isabel F. Cruz and Afsheen Rajendran. 2003. Exploring a new approach to the alignment of ontologies. In *Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, in cooperation with the International Semantic Web Conference*.

Isabel F. Cruz, William Sunna, and Anjli Chaudhry. 2004. Semi-automatic ontology alignment for geospatial data integration. *GIScience*, pages 51–66.

James Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In *Workshop on Unsupervised Lexical Acquisition*, pages 59–67, Philadelphia, PA, USA.

AnHai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, and Alon Halevy. 2003. Learning to match ontologies on the semantic web. *VLDB Journal*, 12(4):303–319.

Jérôme Euzenat, Heiner Stuckenschmidt, and Mikalai Yatskevich. 2005. Introduction to the ontology alignment evaluation 2005. *http://oaei.inrialpes.fr/2005/results/oaei2005.pdf*.

Eduard Hovy. 2002. Comparing sets of semantic relations in ontology. In R. Green, C. A. Bean, and S. H. Myaeng, editors, *Semantics of Relationships: An Interdisciplinary Perspective*, pages 91–110. Kluwer.

T. C. Hughes and B. C. Ashpole. 2005. The semantics of ontology alignment. *Draft Paper, Lockheed Martin Advanced Technology Laboratories, Cherry Hill, NJ. http://www.atl.lmco.com/projects/ontology/ papers/ SOA.pdf*.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284.

Shuang Liu, Clement Yu, and Weiyi Meng. 2005. Word sense disambiguation in queries. In *ACM Conference on Information and Knowledge Management (CIKM2005)*, Bremen, Germany.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain.

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. 1990. Introduction to wordnet: an online lexical database. *International Journal of Lexicography*, 3 (4):235–244.

Vincent Ng and Claire Cardie. 2003. Bootstrapping coreference classifiers with multiple machine learning algorithms. In *The 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*.

Natalya Fridman Noy and Mark A. Musen. 2000. Prompt: Algorithm and tool for automated ontology merging and alignment. In *National Conference on Artificial Intelligence (AAAI)*.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using semantic relatedness for word sense disambiguation. In *Fourth International Conference on Intelligent Text Processing and Computational Linguistics (CiCLING-03)*, Mexico City.

Erhard Rahm and Philip A. Bernstein. 2001. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI-99, Sixteenth National Conference on Artificial Intelligence*.

Rajen Subba and Sadia Masud. 2004. Automatic generation of a thesaurus using wordnet as a means to map concepts. *Tech report, University of Illinois at Chicago*.

Dan Tufis and Oliver Mason. 1998. Tagging romanian texts: a case study for qtag, a language independent probabilistic tagger. In *First International Conference on Language Resources & Evaluation (LREC)*, pages 589–596, Granada, Spain.

Kurt VanLehn, Collin Lynch, Kay Schulze, Joel Shapiro, Robert Shelby, Linwood Taylor, Donald Treacy, Anders Weinstein, and Mary Wintersgill. 2005. The andes physics tutoring system: Five years of evaluations. In *12th International Conference on Artificial Intelligence in Education*, Amsterdam.